Demonstrating Reproducibility Using ICLR 2018 Conference Submissions

Jai Hebel
jai.hebel@mail.mcgill.ca
260744897

Victoria Madge
victoria.madge@mail.mcgill.ca
260789644

Zsombor Balassy
zsombor.balassy@mail.mcgill.ca
260516201

***Abstract*—Reproducibility is a large and ongoing problem in the scientific publishing community, and there are reports of growing issues with credible results due to reproducibility in the field of machine learning and computer science. The purpose of this project is to explore reproducing empirical results from ICLR Conference Submissions. The selected paper, titled, "Lung Tumor Location and Identification with AlexNet and a Custom CNN," (https://openreview.net/forum?id=rJr4kfWCb) uses an AlexNet and a custom-built CNN to classify nodules of CT scans from patients as either benign or cancerous, and compares the results from both classifiers. This report describes the methodologies used to reproduce the empirical results from the paper, and reports on the results which were successfully produced, and discrepancies between those produced and the empirical results from the paper.**

This project is in response to Professor Pineau's Reproducibility Challenge. The project team is signed up under the challenge as 'Team JVZ'. All code used in the reproducibility methodology can be found on GitHub here: https://github.com/vmadge/lungtumour_iclr2018. The Abstract at the end of this report contains the Executive Summary submitted to the authors on OpenReview.

***Index Terms*—Reproducibility challenge, ICLR 2018, Lung tumour, AlexNet, CNN**

I. Introduction

There exists a growing need to reproduce scientific publications as there is evidence of a lack of robustness in their findings. Reproducibility in scientific research is when the results from a previous study are duplicated using the same methods as those used by the original researcher [1]. Reproducibility in scientific research is important to illustrate robustness of the findings and to deem the findings credible. The purpose of this project is to explore reproducibility of empirical results from the International Conference on Learning Representations (ICLR). The goal is to assess if the experiments from the selected paper are reproducible, and to determine if the conclusions of the paper are supported by results found after reimplementing the methodology. The paper under question is titled "Lung Tumor Location and Identification with AlexNet and a Custom CNN." Authors are listed as anonymous as the paper is under double-blind review. A technical summary of the paper, followed by an in-depth description of the reproducibility methodologies utilized, and a presentation, comparison and discussion of empirical results will be described in this report.

II. Technical Summary

The paper titled "Lung Tumor Location and Identification with AlexNet and a Custom CNN" seeks to compare the empirical results from an AlexNet, a popular image identification CNN architecture, to a custom-built Convolutional Neural Network (CNN) [2]. The task was a binary classification problem, determining whether patient scans of lung nodules were benign or cancerous. The authors used the Luna 2016 Challenge dataset, which was derived from the LIDC-IDRI dataset [3]. The dataset includes 888 patients with several CT scans per patient, and included annotations of locations of tumour nodules with a benign or cancerous tag. Annotations were confirmed by at least three radiologists. The dataset of CT scans came in both DICOM and MDH formats, which are common medical image file formats, and were initially processed into PNG format. This was done to condense the image information to what was needed for the AlexNet and CNN without losing image information (as with a JPG file format), and to take advantage of MATLAB's image data storage object, thus reducing memory usage since all images did not have to be imported at the same time. The CNN and AlexNet were compiled in MATLAB 2016b [6] using four NVIDIA GPUs with a total of 32GB memory. Both architectures are described in detail within the paper, both pictorially and descriptively [2]. Figure 3 depicts the custom-built CNN as depicted from the paper.

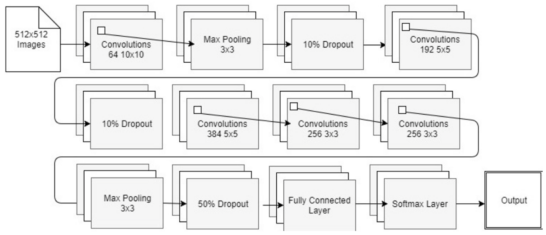


Fig. 1: Custom CNN architecture from [2]

The AlexNet is a famous, highly-flexible CNN architecture used for image identification and is depicted in Figure 2.

The dataset was split into a 70:30 train-test split, with ten pre-divided subsets. Authors evaluated the performance of the CNN and AlexNet using accuracy, precision, recall, F1 score, Matthew's Correlation Coefficient (MCC), False Positive Rate

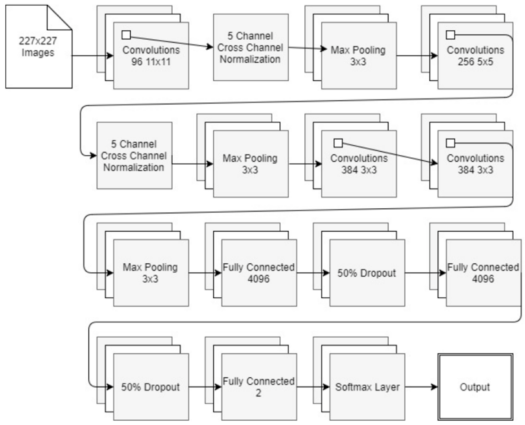


Fig. 2: AlexNet architecture from [2]

(FPR) and False Negative Rate (FNR). Empirical results from the paper will be further discussed in Section IV. From the results, the authors concluded that the custom CNN yielded better performance metrics than the AlexNet.

Authors report future work which includes using the custom CNN to automatically locate the tumours within the patient scans. Authors have already reported progress towards this goal. This includes converting annotations from the 2016 Luna dataset from patient to image coordinates, in order to have the locations of the tumours on the images for training purposes.

III. Reproducibility Methodology

The aim of this project is to reproduce all empirical results reported from the selected paper. The following section will discuss how the results were reproduced in terms of the data pre-processing performed, and the algorithms used.

A. Data Pre-Processing

The dataset specified from the paper was not open to the public through the Luna 2016 Challenge. But since the dataset was derived from the publicly available LIDC-IDRI dataset, it was assumed that the patient subsets were the same. Using the guidelines defined by the Luna Challenge [4], only scans with a slice thickness of no greater than 2mm were used to replicate the Luna 2016 Challenge dataset. Images were therefore downloaded in DICOM format from the LIDC-IDRI website and converted to PNG as per the authors' pre-processing methodology [2]. In the original DICOM format, the size of the dataset was 128GB, and when converted to PNG the file size reduced to 30GB. The final patient subset count was 896 patients, differing from the 888-patient subset described in the paper [2].

The annotations of the dataset were not converted from patient to image coordinates, as was performed in [2], because the AlexNet and CNN are classifying images which contain nodules as either cancerous or benign, and are not locating the nodules within each image (as this is to be done in future work by the authors [2]). Labels for denoting whether nodules were benign or cancerous were appended to the image dataset to be used in both the CNN and AlexNet. This was achieved by indexing the supplied xml label files [5] for each patient, and indexing any image slice names that were labeled as containing a significant nodule.

When attempting to obtain the dataset from the Luna 2016 Challenge, it was noted that the dataset came pre-divided into ten subsets, which explains the method of reporting performance metrics of both the AlexNet and CNN (i.e. Tables 1 and 2 from [2]). The method of dividing the subsets and the number of files included per subset were unknown; however, given the classification time from the results in [2], it was assumed that most datasets were approximately the same size. The dataset acquired from LIDC-IDRI was not previously divided into ten subsets, so the dataset was randomly divided into ten equally sized stratified subsets. Stratifying the subsets ensures that the CNN trains on a well-represented class distribution of the overall dataset. Randomly dividing the subsets was the best possible assumption as to how the subsets were divided, although it is noted that the files included in each subset will differ between the dataset obtained in this project and the one used from the Luna 2016 Challenge by the authors in [2].

Each subset was then further divided into a 70:30 train-test split as per the methodology in [2] and were used to train the AlexNet and CNN. The total number of patients used to train the AlexNet and CNN is 627, leaving 269 patient files for testing purposes. This narrowly differs from the total number of patients used for training and testing as mentioned in [2]. Algorithm selection and hyper-parametrization will be described in the next section.

B. Algorithm Selection

The original code used to produce empirical results from [2] was not available to the project team; however, the architectures for both the CNN and AlexNet were described in the paper.

Starting with the custom CNN, the authors describe both pictorially (as seen in Figure 3) and descriptively (from [2]) the architecture of the custom-built CNN. However, there are discrepancies between the depicted architecture and the step-by-step implementation. All discrepancies between the pictorial and descriptive architecture are listed below:

1) The description mentions a maximum pooling layer of size 2x2, after the convolution layer which uses 192 5x5 convolutions, which is missing from Figure 3
2) The description mentions a convolution layer, using 256 3x3 convolutions, after two convolution layers using 256 3x3 convolutions, which is missing from Figure 3.
3) The description mentions a convolutions layer, using 128 3x3 convolutions, after two convolutions layers using 256 3x3 convolutions, which is missing from Figure 3.
4) The fully connected layer was not mentioned in the description but is shown in Figure 3.
5) The description mentioned that the CNN is fed into the RCNN object detector to locate and identify benign and

cancerous nodules (i.e. tumours), but this is not shown in Figure 3.

Taking these discrepancies in mind, a CNN was built following the architecture of the description, with the added fully connected network as seen from Figure 3. The input layer to the CNN was altered such that the network could run on the available computing infrastructure. Available infrastructure included an NVIDIA Titan X GPU with 12GB of memory. Note that due the advanced computational power of the Titan X, MATLAB 2017a [7] had to be used as the previous version (i.e. MATLAB 2016b) that the authors used does not support this GPU [8]. The authors report using four GPUs with 8GB of memory each for a total of 32GB. Given the lack of computation power - specifically, the available GPU memory - images were down-sampled from 512x512 to 64x64. The mini-batch size, number of epochs, and learning rate for the CNN were not specified in [2], therefore the default values provided by MATLAB's NeuralNet Training Options were used or modified [9]. Mini-batch size was an automatic default size of 128, the number of epochs used was 20, and the learning rate was kept at the default 0.001. Performance metrics, including accuracy, precision, recall, F1 score, Matthew's Correlation Coefficient (MCC), False Positive Rate (FPR) and False Negative Rate (FNR) were generated for each subset. Empirical results for each subset will be compared to the results obtained from the reproducibility methodology in Section IV.

The AlexNet is a popular image identification package from [10]. The available pretrained AlexNet architecture is standard throughout most online available packages, including the MATLAB package obtained to run in version 2017a [11]. The architecture from the package used is therefore the same as the one described in the paper (see Figure 2) with an image input size of 227x227 etc. Although there were discrepancies between what was written in the paper and what was depicted in Figure 2 from [2], it was assumed that the description of the AlexNet from the paper was simply a brief overview of the architecture, as it was not as detailed as Figure 2. All hyper-parameters which were specified in the available package were used, since no hyper-parameters were specified for the AlexNet in [2], as was the case with the custom CNN. A mini-batch size of 250 was used, number of epochs was 30, the initial learning rate was 0.00125 with a learning rate drop factor of 0.1 and a learning rate drop period of 20. The same computing infrastructure was available to run the AlexNet (i.e. an NVIDIA Titan X with 12GB of memory). The AlexNet ran at full image resolution (227x227) despite the computation power being reduced from 32GB to 12GB of available memory on one GPU versus four GPUs. The same performance metrics were generated for the AlexNet as those for the CNN. Empirical results for each subset in [2] will be compared to the results obtained from the reproducibility methodology in section IV.

IV. Results

This section will compare the empirical results from [2], and the reimplementation results.

Tables I and II show the accuracy, precision and recall for the custom-built CNN from [2], and the reimplementation, respectively. From Table II, it can be seen that the accuracy did not change between subsets as with the empirical results in [2]. The reimplemented CNN compiled at the down-sampled image size of 64x64 only yielded results which simply predicted the majority class. This resulted in a recall of 0, and an incalculable precision since there were no true positives or false positives (i.e. everything was predicted to be negative, given this was the majority class).

TABLE I: Performance metrics for CNN classifier from [2]

Subset	Accuracy	Precision	Recall
0	99.80%	100.00%	99.80%
1	99.76%	100.00%	99.76%
2	99.83%	100.00%	99.83%
3	99.81%	100.00%	99.81%
4	99.85%	100.00%	99.85%
5	99.70%	100.00%	99.70%
6	99.86%	100.00%	99.86%
7	99.73%	100.00%	99.73%
8	99.76%	100.00%	99.76%
9	99.74%	100.00%	99.74%

TABLE II: Performance metrics for CNN classifier using the reimplementation methodology described in Section III

Subset	Accuracy	Precision	Recall
0	92.65%	NaN%	0.00%
1	92.65%	NaN%	0.00%
2	92.65%	NaN%	0.00%
3	92.65%	NaN%	0.00%
4	92.65%	NaN%	0.00%
5	92.65%	NaN%	0.00%
6	92.65%	NaN%	0.00%
7	92.65%	NaN%	0.00%
8	92.65%	NaN%	0.00%
9	92.65%	NaN%	0.00%

Tables III and IV show the accuracy, precision and recall for the Alexnet from [2], and the reimplementation, respectively. From Table IV, it is noted that accuracy, precision, and recall are all lower than the empirical results from Table III.

Tables V and VI show the comparisons between all performance metrics for both the AlexNet and CNN from [2] and from reimplementation results. The False Positive Rate and False negative Rate from Table VI for the reimplemented CNN affirms the hypothesis that the network only predicted the majority class (i.e. benign). Both the reimplemented AlexNet and the CNN classifiers performed worse than the classifiers from [2]. However, the classification time of the CNN, in both cases, was better than the classification time of the AlexNet. Reasonings behind the discrepancies noted between the reimplemented results and the empirical results will be discussed in Section V.

TABLE III: Performance metrics for AlexNet from [2]

Subset	Accuracy	Precision	Recall
0	99.69%	99.84%	99.84%
1	99.76%	99.98%	99.77%
2	99.77%	99.90%	99.86%
3	99.78%	99.93%	99.85%
4	99.79%	99.91%	99.88%
5	99.44%	99.68%	99.76%
6	99.87%	100.00%	99.87%
7	99.55%	99.79%	99.76%
8	99.60%	99.82%	99.77%
9	99.62%	99.83%	99.79%

TABLE IV: Performance metrics for AlexNet classifier using the reimplementation methodology described in Section III

Subset	Accuracy	Precision	Recall
0	85.95%	23.45%	40.32%
1	90.75%	36.15%	33.79%
2	90.90%	35.28%	28.66%
3	91.55%	39.89%	29.64%
4	86.33%	25.51%	44.86%
5	92.41%	46.53%	22.53%
6	90.03%	32.29%	32.61%
7	89.75%	30.69%	31.42%
8	91.32%	35.26%	21.74%
9	87.18%	26.41%	41.70%

V. Discussion

Discrepancies between the empirical results reported in [2] and the reimplementation results can be attributed to clarity of the paper, availability and partitioning of the dataset, availability of the code from the paper, alignment between the code described and depicted in the paper, availability of hyper-parameters used, differences in computing infrastructure (i.e. memory requirements fo the GPUs used), reimplementation effort, and communication with the authors.

A. Clarity of the paper

Apart from the technical challenges faced to reproduce the methods of the paper, there were challenges with interpreting the paper itself. As no members of the project team are part of the scientific field that specializes in the analysis of CT scans, some observations might be unfounded. However, a lack of consistency and scientific rigour was observed regarding the communication of experimental findings to a broader scientific community. The first observation regards the title of the paper ("Lung Tumor Location and Identification with AlexNet and a Custom CNN") which highlights to two goals that are assumed to be addressed within the paper: lung tumour location, as well as identification. However, the machine learning techniques implemented were for identification purposes only, and though particular sections made the reader think that the authors had implemented tumour location (e.g. the last step in the detailed description of their architecture, see also Section III), looking at the capabilities of the CNN model and the results section from [2] the conclusion can be drawn that this goal was not reached. Later in the conclusion section, the authors state that lung tumour location is indeed future work. Given this, the title on its own misleading as it did not represent the goals achieved in the paper, but rather stated a goal that the authors consider future work. This observation will be further discussed in sub-Section V-F.

TABLE V: Comparison of AlexNet and custom CNN from [2] to AlexNet and custom CNN from reproducible methodology

Network	Accuracy	Precision	Recall	Train Time
AlexNet from [2]	99.72%	99.93%	99.85%	19.60h
CNN from [2]	99.79%	100.0%	99.85%	1.61h
AlexNet	89.62%	30.64%	32.73%	25.47h
CNN	92.65%	NaN%	0.00%	22.62h

TABLE VI: Comparison of AlexNet and custom CNN from [2] to AlexNet and custom CNN from reproducible methodology

Network	FPR	FNR	F1 Score	MCC
AlexNet from [2]	54.26%	0.15%	0.99	0.37
CNN from [2]	0.00%	0.15%	0.99	0.56
AlexNet	6.00%	67.00%	0.32	0.26
CNN	0.00%	100.00%	0.99	0.56

Another observation of a lack of scientific rigour from the paper pertains to the motivation of the paper, which is a dataset challenge, where both the input and the output data were given to participants in a particular format. Citing such a challenge would be sufficient in properly describing the dataset format, although a better approach might have been to describe the input and output format with a detailed procedure on how the dataset was actually obtained. The main concern was how the input formats mentioned throughout the paper were ambiguous. Looking at the architecture of the CNN, it was expected that the inputs would be single images, as opposed to a 3D stack of images which the original dataset provides per patient. With the former, the output would be a cancerous or benign label for each image, whereas for the latter, it would be a single label per patient. However, as opposed to what the described architecture implies, the input is referred to multiple times as "sets of scans" and, simultaneously, the total number of inputs equals to the actual number of patients that were used in the study. Based on the previously mentioned information, this could not be true as there were more than 100 scans per patients, which makes the number of inputs way more that what they state. Because of this, a single label per image was used due to the specified architecture.

The paper also uses terms (e.g. smoothed, unsmoothed images) that were not defined previously such that the reader could understand the terminology, abbreviations were defined after their initial usage (e.g. RCNN for Regions with CNN, was used prior), and the paper submission also contained grammatical errors. All of these miss-clarifications from the paper introduced confusion as to how to properly reproduce empirical results. It is suggested that authors utilize proofread-

ing, either by a third party or by the authors themselves, which may have helped prevent ambiguity in the methodology.

B. Availability and partitioning of the dataset

Given the original dataset from the Luna 2016 Challenge was not available to the public, the dataset was obtained from LIDC-IDRI directly. Since the Luna 2016 Challenge dataset was derived from the LIDC-IDRI dataset, an assumption was made that, should the same slice thickness constraints be applied as the Luna 2016 Challenged used, the datasets would be identical. However, this was not the case. The resulting dataset obtained from LIDC-IDRI contained 896 patient subsets, differing from the 888 patient subsets reported from [2]. A larger patient cohort, although not significantly larger, could have resulted in higher empirical results obtained from the reimplementation of the CNN and AlexNet since larger datasets typically reduce Type II error; but given other discrepancies between the methodologies, this effect was not observed. Had it been observed, it would not have been a significant increase in performance metrics as the dataset was not significantly larger than the dataset from the Luna 2016 Challenge. The Luna 2016 Challenge dataset also came pre-partitioned into 10 subsets, of which the files contained within each subset were unknown, and the partitioning methods were unknown as well. The dataset (896 patients) from this project was divided randomly into 10 stratified subsets, assuming this was the strategy employed by the creators of the Luna 2016 Challenge dataset. It is noted that the subsets created by the Luna 2016 Challenge and the subsets created for the purpose of reproducing the results from [2] are not the same and can therefore account for some discrepancies between the empirical results. However, if all subsets are stratified such that the class distributions are well-represented in the subsets, the discrepancies in the order of patients included in the subsets should not have a significant impact.

C. Availability and alignment of code

Given that the code was not available, the CNN and AlexNet were obtained through MATLAB packages and some architecture manipulation was conducted to match the architecture from [2]. Although the AlexNet package shares the same architecture as that of the paper, and architecture for the custom-built CNN was well-described, there may have been some optimization used in the CNN or AlexNet which would have allowed for faster computation, closing the gap between the reimplemented computation time and the computation time noted in the empirical results from the paper. There also existed some differences in the CNN architecture described from Figure 3 and from the text in the paper which were discussed in Section III. The ambiguity in which architecture was the true architecture used to produce the empirical results may have a large significance on result discrepancies.

D. Availability of hyper-parameters

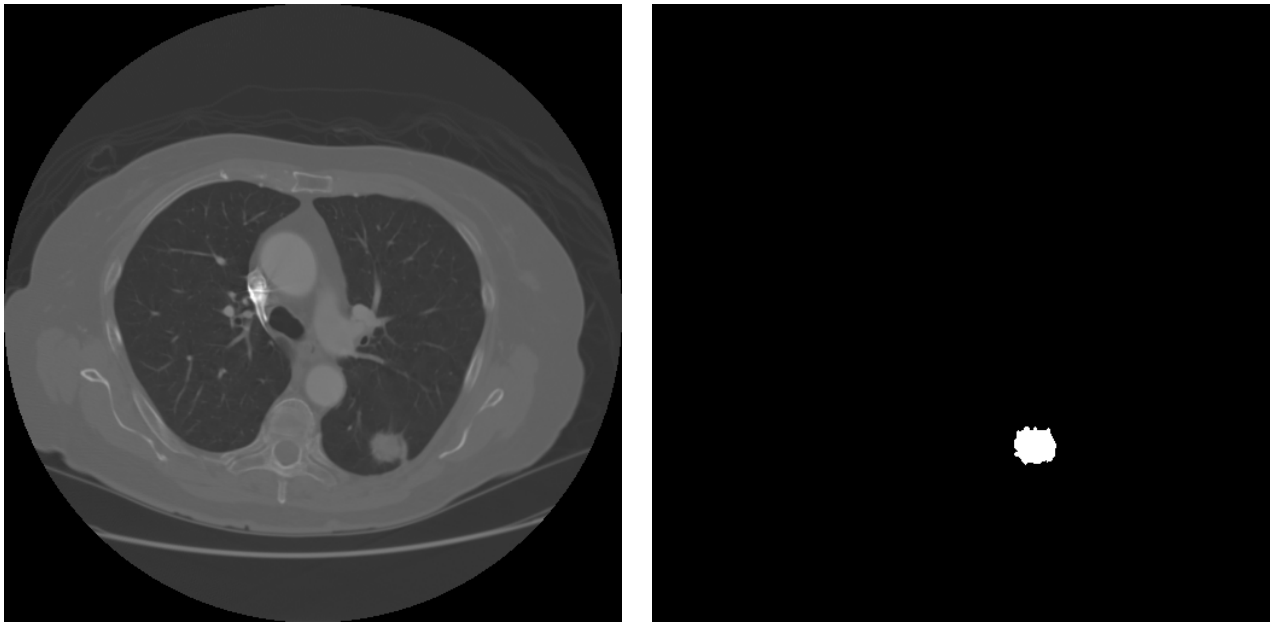
The unavailability of hyper-parameters, including learning rate, mini-batch size, and number of epochs for the custom-built CNN may have contributed to the differences seen in results. While such parameters can be optimized with extensive cross-validation, the large dataset size and lengthy computational time (as well as the lack of adequate computational infrastructure) made such extensive optimization difficult for the team to implement. Ultimately, it cannot be demonstrated that the reported success of the custom CNN was not superior to the AlexNet owing to better hyper-parameter optimization by the authors.

E. Memory requirements

The most significant impediments to accurate replication of the reported results were the nature of the learning architecture, the inherent computational requirements of said architecture, and the resources available to the project team. Initial attempts to train the CNN showed that calculations of the final layers in the architecture required construction of a tensor, requiring over 24GB of available memory. These memory requirements were present despite the decision to use MATLAB Datastore objects to only load images in the current mini-batch into working memory, and persisted despite attempts to reduce mini-batch size. The project team only had one NVIDIA Titan X GPU with 12GB memory, which was an insufficient amount of memory for the algorithm to process. In contrast, the authors from [2] used 4 NVIDIA GPUs with 8GB of memory each, and so were capable of handling up to 32GB of calculations for the custom CNN.

Given the memory constraints, the decision was made to down-sample the 512x512 pixel PNG images in order to fit the problem into the 12GB GPU. Various resolutions were attempted, and the maximum resolution that allowed the algorithm to converge was found to be 64x64 pixels (1/8th of the original resolution). PNG images were thus resized as they were read from the Datastore object in each mini-batch. Given that some nodules were labeled as less than 8 pixels in diameter, and appeared to range up to only about 20 pixels in diameter, the necessity to reduce image resolution and the subsequent data loss no doubt had a significant effect on the performance of the CNN. With such a low resolution, and without using pixel labels to train, nodules were likely indistinguishable by the CNN, resulting in the majority-class prediction shown in Section IV.

While attempts to re-calculate the CNN at full resolution were pursued, none were successfully completed before the publication of this report. Aside from attempts to access a superior GPU, the team was recommended by the course's TA to forgo replicating the results in MATLAB, and attempt to utilize Python libraries. Given that Python deep learning libraries may require less memory to compute than MATLAB toolboxes, this seemed a promising alternative. However, the team was unable to translate the existing MATLAB code into Python in a manner that allowed for the same mini-batch loading of images as the Datastore object, and were unable to load the entire PNG library into working memory at once. Future efforts would primarily consist of continuing to pursue implementation of the custom CNN using the PyTorch library [12] to minimize memory requirements; however, the large



(a) (b)

Fig. 3: A sample CT slice demonstrating a) the scaled PNG image of the slice and b) the corresponding pixel mask derived from the ground truth labels.

memory requirements of such an algorithm do speak to the accessibility and the reproducibility of the algorithm. Such a large graphics-memory requirement may limit the usability of the algorithm, as demonstrated by the team's difficulties, and is certainly a cost requirement to consider when comparing the proposed architecture against baseline or alternative classifiers.

F. Pixel labels

It is worth noting that, while the authors of the paper did access and calculate pixel labels for each CT slice, these pixel labels did not appear to be used in the learning architecture. The AlexNet baseline that was used is a classification model, and was previously trained on a dataset with 1000 object labels before being retrained on the CT images [11]. Such a model is not designed for identification, neither for segmentation, and similarly the custom CNN architecture is representative of a binary classification problem. It seems contradictory that the pixel labels were not used when the authors detail their calculation and translation from world-coordinates to pixel coordinates [2]. However, if the pixel labels were somehow implemented into either the AlexNet or CNN architecture, it was not detailed in the report.

The inclusion of such information would likely improve the quality of tumour identification, as pixel labels provide the networks with a more specific description of what to look for in the images. Such a methodology would allow for a full segmentation of each image slice into benign and cancerous regions, and slices could be identified as benign in the absence of any cancerous pixels. By converting the *.xml label files into label masks for each CT slice (see Figure 3), a custom CNN that classifies each pixel and outputs a prediction mask is possible. The inclusion of pixel labels may account for the discrepancy between the reported and reproduced results for the AlexNet; however, such an algorithm would require a modified AlexNet, as the AlexNet produces object recognition labels for the entire image, and not labels for individual pixels. Additional future work by the team would be to implement such pixel labeling for image segmentation in addition to the demonstrated tumour identification.

G. Reimplementation effort

The reimplementation effort of the project team as a whole was challenging given the large dataset, making data manipulation computationally expensive and thus time consuming. Better knowledge on GPU infrastructure would have yielded a faster or a better results outcome as well. Given that the dataset constituted of 128GB of CT images, and training for each algorithm exceeded 22 hours, cross-validation and adjusting the methodology after testing to completion were not feasible. Particular expertise in working with large image datasets in Matlab's Neural Network Toolbox, as well as thorough documentation of the methodology by the previous authors, would have allowed the team to better predict the difficulties incumbent in such a project.

H. Communication with authors

Communication with the authors was attempted on OpenReview from a number of entries outside of the project team. It is, however, noted that more contact with the authors may have yielded better results if authors agreed to give code, and insight on CNN architecture, etc.

VI. Conclusion

To summarize, the empirical results from the selected paper titled, "Lung Tumor Location and Identification with AlexNet and a Custom CNN" did not match the results produced from an attempted reproducibility methodology followed for the Reproducibility Challenge from course COMP 551. Architectures of the CNNs involved (i.e. the custom CNN and AlexNet) did not yield enough information to accurately reproduce the same results. The reimplemented CNN yielded results in which only the majority class (i.e. benign) was predicted (approximately 92%), and AlexNet results were lower than those expected from [2]. Although the conclusion from [2] that the custom CNN performs better than the AlexNet does hold true in the results which were produced in this report, most empirical results were unable to be properly reproduced. The most significant contributor to the differences in results observed is the computation power, which ultimately affected overall performance of the classifiers, specifically the CNN which had a reduced resolution of the input layer. Given the challenges in reproducing exact outcomes from [2], specifically given the memory requirements, there is not enough evidence to conclude that the empirical results from the paper can be reproduced. Given ample GPU memory, empirical results may have been matched, but future work to improve memory requirements could only reveal this as truth. Suggestions for the authors are to include detailed and unambiguous dataset retrieval and description, as well as detailed and uncontradicting descriptions of the network architectures of the various classifiers used, and to use a title that aptly describes the work performed in the paper.

VII. Acknowledgments

Special thanks to Koustuv Sinha for his patience and guidance in helping us set up a GPU instance to run the Python CNN (even though this was unsuccessful from the team's end), and thanks to Professor Pineau for her patience, advice, and understanding. A special thanks to Dr. Louis Collins and Vlad Fonov for setting up and allowing us to use their Titan X GPU from the NIST lab.

VIII. Statement of Contributions

Below are the project contributions from each author.

Jai Hebel: I led the data pre-processing and manipulation of input images and output labels, assisted in implementing CNN and AlexNet architectures in MATLAB, as well as lead the Python implementation of the CNN and AlexNet architectures and contributed to report writing.

Victoria Madge: I contributed to the majority of the report writing, and produced results for both the CNN and AlexNet architecture in MATLAB. I also contributed to the spotlight presentation, and executive summary.

Zsombor Balassy: I contributed to the data pre-processing and manipulation of input images and output labels, assisted in implementing CNN and AlexNet architectures in MATLAB, as well as contributed to the spotlight presentation, report writing, and lead the executive summary write up.

With this, we hereby state that all the work reimplemented from the ICLR 2018 conference submission, which is presented in this report, is that of the authors.

References

[1] K. Bollen, J. T. Cacioppo, R. M. Kaplan, and J. A. Krosnick, "Social, Behavioral, and Economic Sciences Perspectives on Robust and Reliable Science," 2015.

[2] Anonymous, "Lung Tumor Location and Identification with AlexNet and a Custom CNN," *ICLR 2018*, pp. 1-7, 2017.

[3] "LIDC-IDRI", *Cancer Imaging Archive (TCIA)*, 2014. [Online]. Available: https://wiki.cancerimagingarchive.net/display/Public/LIDC-IDRI. [Accessed: 15- Dec- 2017].

[4] C. Jacobs and B. van Ginneken, "LUNA16 - Home", *Luna16.grand-challenge.org*, 2017. [Online]. Available: https://luna16.grand-challenge.org/.

[5] Wouter Falkena, "xml2struct", *Mathworks*, 2010. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/28518-xml2struct.

[6] Mathworks. Matlab 2016b. Mathworks Inc., Natick, MA, 2016.

[7] Mathworks. Matlab 2017a. Mathworks Inc., Natick, MA, 2017.

[8] "Bug Reports", Mathworks.com, 2017. [Online]. Available: https://www.mathworks.com/support/bugreports/1439741.

[9] MathWorks Inc., "trainingOptions: Options for training neural network," 2017. [Online]. Available: https://www.mathworks.com/help/nnet/ref/trainingoptions.html.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks." *Advances in neural information processing systems*, pp. 1097-1105, 2015

[11] MathWorks Neural Network Toolbox Team, "Neural Network Toolbox(TM) Model for AlexNet Network," 2017. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/59133-neural-network-toolbox-tm–model-for-alexnet-network

[12] Paszke, Adam, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. "Automatic differentiation in PyTorch." (2017).

APPENDIX

Executive Summary:

As part of the Reproducibility Challenge from the course COMP 551 at McGill University, we have attempted to reproduce the empirical results from the paper titled "Lung Tumor Location and Identification with AlexNet and a Custom CNN". Here, we describe a summary of the main challenges that we faced in doing so, as well as the results that we obtained.

The motivation for the paper was a challenge to improve on the classification of CT scans whether they contain cancerous or benign tumors. The publicly available and well-labeled dataset that the paper uses is the modified LIDC-IDRI dataset that contains sets of CT scans of patients with lung tumours. The size of the dataset is 128GB of DICOM images, and is 32GB when converted to PNG. Manipulation of this data was difficult and time consuming on its own.

The authors obtained their input and output sets from the closed Luna 2016 Challenge, which came in a preprocessed format and also came divided into 10 different subsets. We did not have access to the Luna 2016 Challenge dataset (no public access), so we ended up using the publicly available LIDC-IDRI dataset and pre-processed it in a way that matched the Luna dataset description. The authors state that they worked with 888 patients, whereas for our dataset, following the same procedure, yielded 896 patients. This discrepancy in input was not so big that we would expect significantly different results. The only implication is that some additional patient files were removed before finalizing the Luna dataset in a manner that was not documented. Converting the data to PNG files was needed as the original format (DICOM and MDH) contains more than just the pictures of the scans, which we were able to achieve. We also had to extract the output labels from the public dataset ourselves, which could have also contributed to some differences from the data that the paper uses. Given that the authors of the paper had to convert the pixel labels from millimeters to pixels, we believe the labels provided to the authors differed in format from those supplied with the LIDC-IDRI.

It was originally unclear whether we followed the right input and output format on whether a tumour, being cancerous or benign, can be interpreted either as on a per image basis or on a per patient basis. The architectures clearly supports a per image classification, however, some of the descriptions of the paper ambiguously implied otherwise. With the publicly unavailable dataset, such an elaborate pre-processing method, as well as the ambiguity in both the inputs and the outputs, providing the code would have made our reproducibility easier.

The paper implemented two CNN architectures: an AlexNet modified to binary output, and a custom CNN designed by the authors for the purpose of tumour identification. The architecture was described using both figures and a layer-wise text description, however these descriptions were inconsistent. We used a version that contained all the layers from both architectures, which adds another uncertainty in the reproduction of the results. The motivation for using the custom CNN was the ability to decrease the false positive rates compared to the baseline AlexNet. Ultimately, our findings were unable to support or deny this assertion. In our efforts to trying to run both of these models, the difficulty that we faced (apart from the ambiguity in the network architecture) was the lack of proper hardware available to us. We were able to use a NVIDIA Titan X GPU, which only provided 12GB of memory that turned out to be a limiting factor for us as we needed to decrease the resolution of our input images for the custom CNN. We would also like to note the lack of the hyper-parameters identified in the paper of which were used to train the two classifiers.

On a more general note, we had difficulty with the clarity of the paper as some statements were made without antecedent (e.g. smoothed and unsmoothed images), or that some parts were included in the results section that were part of the future work as was clearly stated later on. We suggest having a third party reviewer go through the paper as these nuances made the paper difficult to read.

Please find attached our complete review here: https://github.com/vmadge/lungtumour_iclr2018.